*Design Specification*

**MeetnLunch**

|  |  |
| --- | --- |
| Description of the document |  |
| Title | Design Specification |
| Date | 26/04/2017 |
| Author(s) | *50161375* ***KAUCH Jonathan***  *50161378* ***TOMA Ivan***  *50161387* ***FRAYSSE Christophe***  *50161388* ***LIN Chaohui***  *50161390* ***MOUTIER Baptiste***  *50161389* ***ZABANDITH Brian*** |
| Subject | <Product Sheet of design specification of MeetnLunch Mobile Application> |
| Version | 1.0 |

Tableau de Révisions :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Auteur | Section(s) | Commentaire |
| 26/04/2017 | 1.0 | Ivan TOMA  Jonathan KAUCH  Christophe FRAYSSE  Chaohui LIN  Baptiste MOUTIER  Brian ZABANDITH | N/A | Création du document |

Summary

Design Specification

MeetnLunch

.........................................

1 **Architecture descriptions**

................................................................

.

1.1 System configuration

1.2.1 System Diagram

1.2.2 Flow Diagram

1.2.3 Data Description

.................................................................

2 **Process management**

..................................................................

.

2.1 Risk Assessment

2.2 Test Plan

2.3.1 Project Schedule

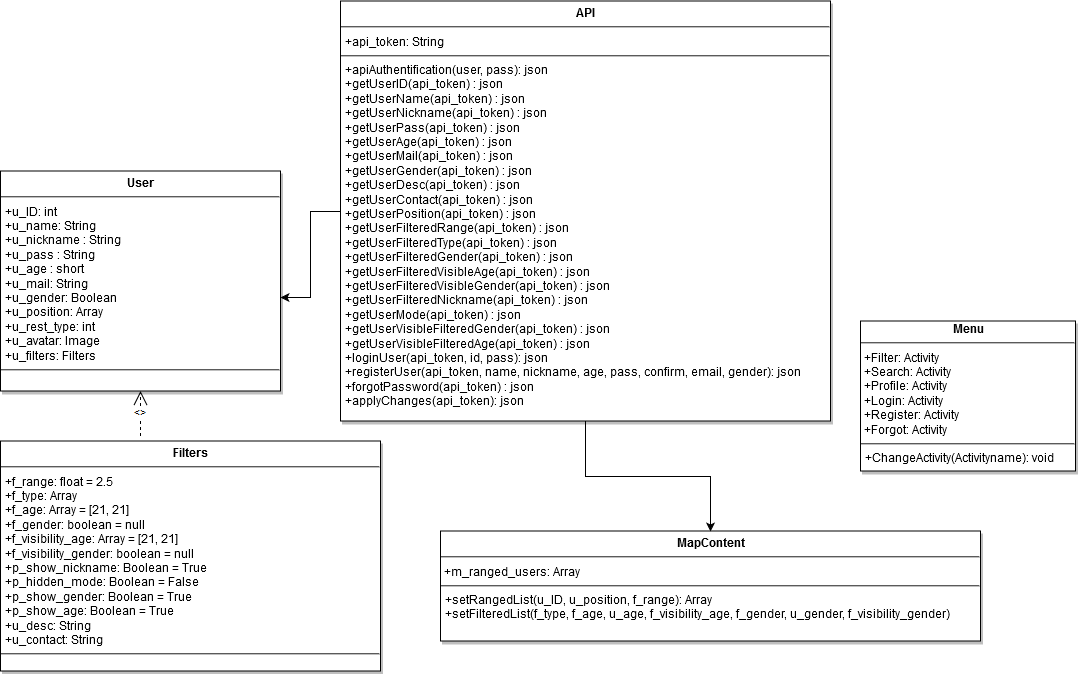
2.3.2 Role Distribution

1. Architecture description

1.1 System configuration

Our application will work with a website and a mobile application (android). Both of them will be connected by an API (Application Programming Interface). The API and the website will be develop with Javascript framework NodeJS (<https://nodejs.org/en/>). And the mobile application will be obviously be develop with Java Android (<https://www.android.com/>).

|  |  |
| --- | --- |
| Section | Technology |
| Back-end | NodeJS, ExpressJS |
| Front-end (webapp) | NodeJS, ExpressJS, HTML, CSS, JQuery |
| Front-end (mobile) | Java Android |
| Server | Heroku |

1.2.1 System Diagram

1.2.2 Flow Diagram

This flow diagram represent our main feature, the matching system. The user will send a request to find a partner, the API will return to the front-end the list of the user available to get lunch. The user will select one user and an invitation will be send to the potential partner. He can accepts or rejects the invitation.



1.2.3 Data Description

**User**

|  |  |  |
| --- | --- | --- |
| **Data** | **Type / Range** | **Description** |
| u\_ID | int | Unique ID per user |
| u\_name | String : 25 | Real name |
| u\_nickname | String : 25 | Nickname |
| u\_pass | String : 20 | Password |
| u\_age | short : 18 - 50+ | Age |
| u\_mail | String | Email Address |
| u\_gender | Boolean | Gender 0 = Male 1 = Female |
| u\_position | Array | An array containing the coordinates of the user, to locate him on Google maps |
| u\_rest\_type | int | Type of restaurant where the user is, defined by the the user |
| u\_avatar | Image | Image uploaded by the user, if none, nothing is displayed |
| u\_filters | Filters | An object containing the applied filters by the user |

**Filters**

|  |  |  |
| --- | --- | --- |
| **Data** | **Type** | **Description** |
| f\_range | Float : 0 - 5 | Distance in km for search |
| f\_type | Array | Type of restaurant |
| f\_age | Array [18 - 50+ , 18 - 50+] | Range of age the user looking for |
| f\_gender | boolean : null, 0, 1 | Gender 0 = Male, 1 Female, null = no preference |
| f\_visibility\_age | Array [18 - 50+ , 18 - 50+] | Range of age the user is visible for |
| f\_visibility\_gender | Boolean | 0 = Not visible 1 = Visible |
| f\_show\_nickname | Boolean | 0 = Not visible 1 = Visible |
| f\_hidden\_mode | Boolean | 0 = Visible 1 = Invisible |
| f\_show\_gender | Boolean | 0 = Not visible 1 = Visible |
| f\_show\_age | Boolean | 0 = Not visible 1 = Visible |
| u\_desc | String : 250 | Description field for the user |
| u\_contact | String : 100 | Contact Informations |

**API**

|  |  |  |
| --- | --- | --- |
| **Data** | **Type** | **Description** |
| api\_token | String | Token returned after logged in the API, mandatory to make requests |

**MapContent**

|  |  |  |
| --- | --- | --- |
| **Data** | **Type** | **Description** |
| m\_ranged\_users | Array | An array of all displayed Users |

**Filters**

|  |  |  |
| --- | --- | --- |
| **Data** | **Type** | **Description** |
| Filter | Activity | FilterView |
| Search | Activity | MapView |
| Profile | Activity | ProfileView |
| Login | Activity | LoginPage |
| Register | Activity | RegisterPage |
| Forgot | Activity | Forgot password page |

2 Process management

2.1 Risk assessment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hazard** | **Person who may be harmed** | **Risk control that will be in place** | **Risk Assessment**  **LOW, MODERATE,**  **HIGH, EXTREME** | **Further action required to control risk** |
| Bad encounter | User | Advice like “Always meet other people in public places” | MODERATE | More demanding identification control (phone number) |
| Sexual harassment | User | No personal information will be displayed on the application (phone number, name, etc) | MODERATE | Limit search to a certain gender |

2.2 Test Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System Classification** | **Functionality** | **Feature description** | **Success result**  **Status code:**  **200 : success**  **All results is in Json format.** | **Fail result**  **Status code:**  **400: errors,**  **401: unauthorized**  **403: Forbidden** |
| API | Connection | Login | 200,  Return user informations | 400, |
| Sign up | 200,  Return created user informations | 400 |
| Forgot password | 200,  Return success message | 400 |
| Profil | Check profil | 200,  Return profil informations | 400 or 401 if user isn’t log in |
| Modify profile informations | 200,  Return new profile informations | 400 or 403 if the profile doesn’t match with current user |
| Partner searching | Searching partner | 200,  Return success message | 400 |
| Accept Partner | 200,  Return partner informations | 400, 401 |
| Reject Partner | 200 | 400 |
| MOBILE | Connection | Login | Access to profile screen | Return to login screen |
| Sign up | Access to login screen | Return to sign up screen |
| Forgot password | Access to login screen | Return to forgot password screen |
| Profil | Check profil | Profile screen | Login screen if user missing |
| Modify profile informations | Profile screen | profile modification screen with errors message |
| Partner Searching | Searching partner | Profile screen | Searching screen with errors message |
| Accept partner | Profile screen | Profile screen with errors message |
| Reject partner | Profile screen | Profile screen with errors message |

2.3.1 Project schedule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Description | | | 04.26 - 05.10 | 05.10 - 05.24 | 05.24 - ~ |
| Design system architecture | | Idea develop | All |  |  |
| Service design | All |  |  |
| Service scenario | All |  |  |
| Use case | Designer |  |  |
| Testing plan | Manager |  |  |
| Scrum | Select | Scrum meeting | All | All |  |
| Add component | Manager, Designer | Manager,  Designer |  |
| Documentation | All | All | All |
| Develop | Coding | Developer | Developer | Developer |
| Refactoring | Developer | Developer | Developer |
| Review | Client meeting | Manager | Manager | Manager |
| Burn chart | Manager | Manager | Manager |
| Project closure | | Wraps up |  |  | All |
| Documentation |  |  | All |

2.3.2 Role distribution

|  |  |
| --- | --- |
| Name | Role |
| KAUCH Jonathan | Designer, Developer |
| FRAYSSE Christophe | Developer |
| LIN Chaohui | Developer |
| TOMA Ivan | Project Manager |
| MOUTIER Baptiste | Developer |
| ZABANDITH Brian | Developer |